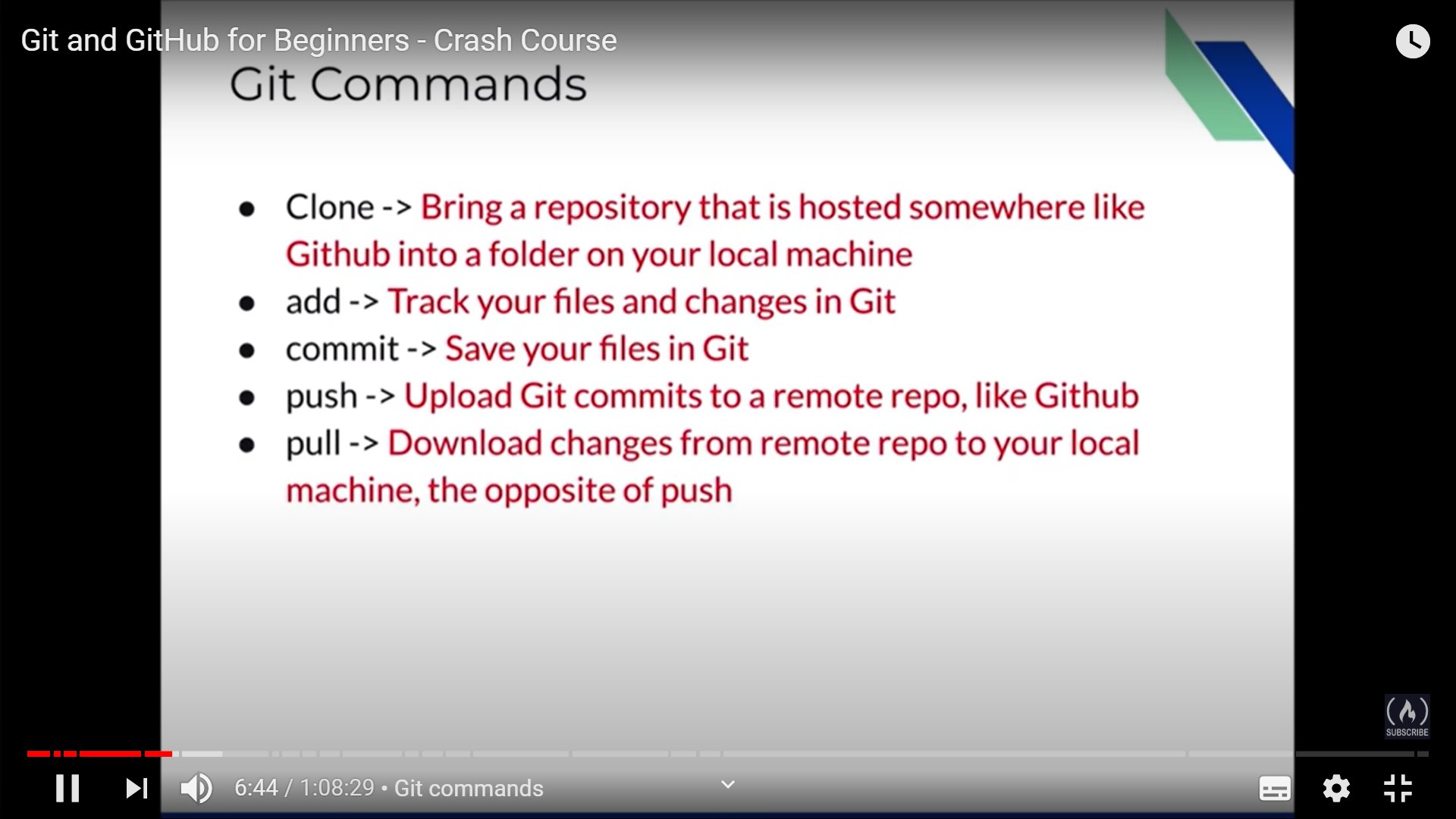
GIT ( Global Information Tracker )

Link :: <https://www.youtube.com/watch?v=RGOj5yH7evk>

Document Link :: <https://git-scm.com/book/en/v2/Git-Basics-Working-with-Remotes>





How to know git version :: git –version

To initialize the git repository, use command : git init

Or git init <ProjectName>

Command : tree .git

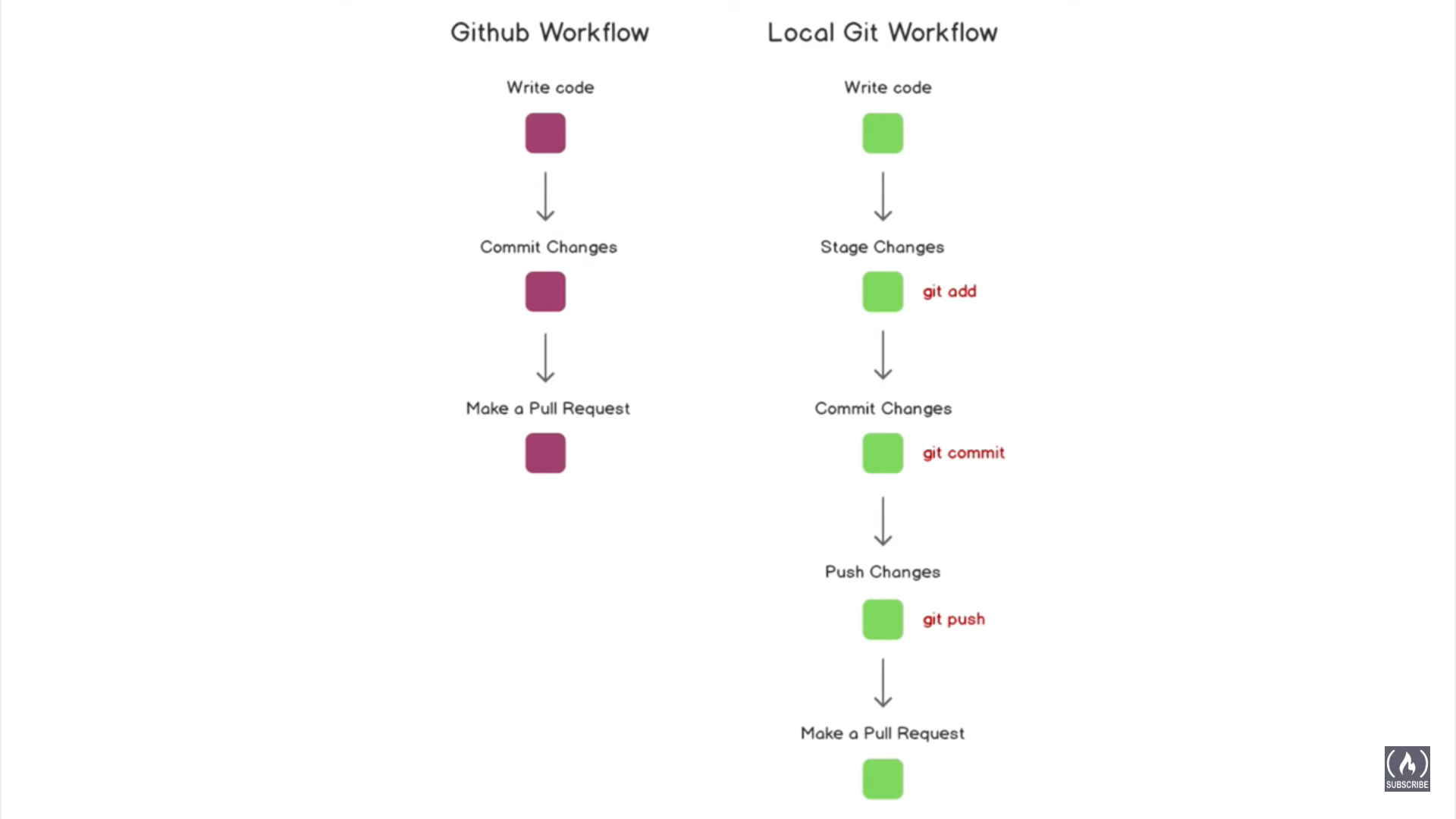
Clone means bring a repository that is hosted somewhere in github into a folder on your local machine along with log history

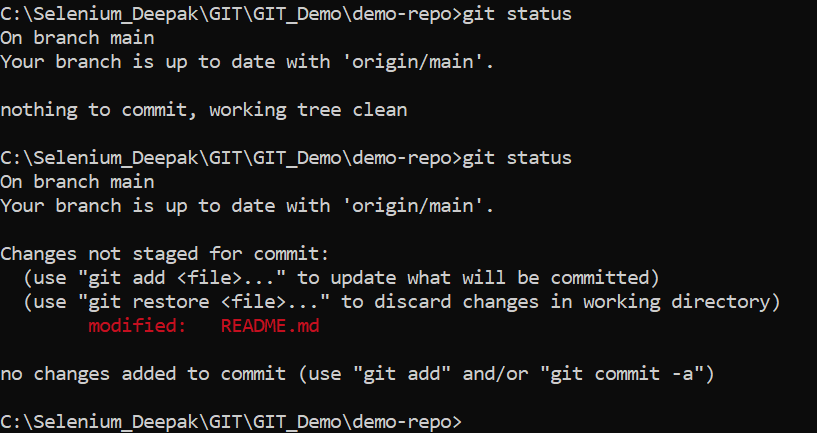
Syntax : git clone <URL>

Clone command : git clone <https://github.com/dmishra4/demo-repo.git>

Status command : To know the git current status

Status Command : git status

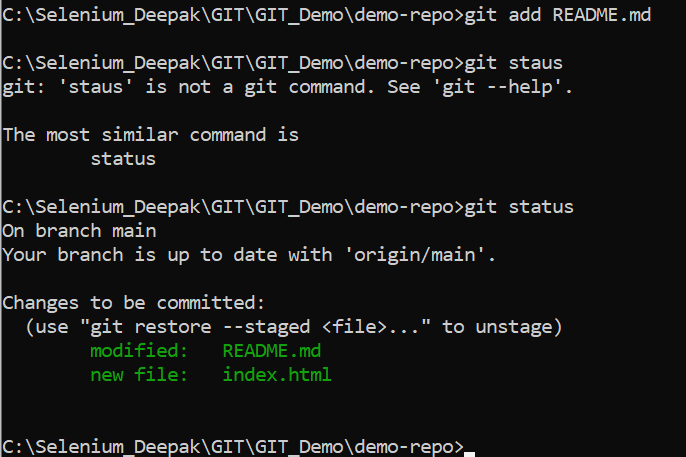




How to add files to track and changes for staged

Git command : git add index.html ( Any particular file )

Add file . ( All file which are untracked or modified )



How to commit in git ::

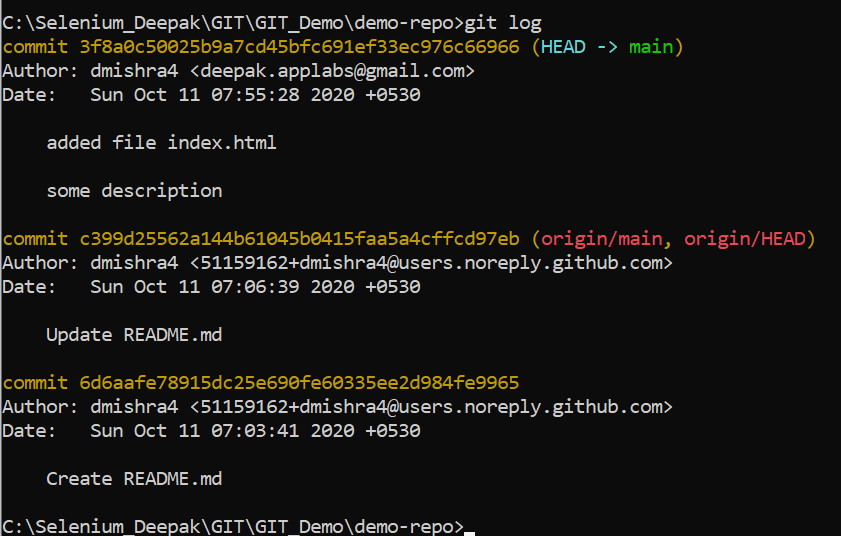
Git command : git commit -m "added file index.html" -m "some description"

-m stand for message and second -m is for description.

################ How to see git log ##################

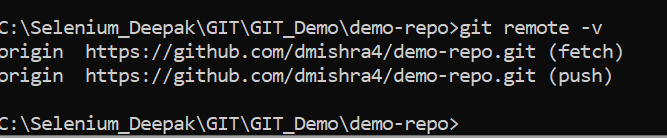
Command :: git log

One more command : git log --oneline --decorate --all --graph



################ How to see whether git is connect to remote or not ##################

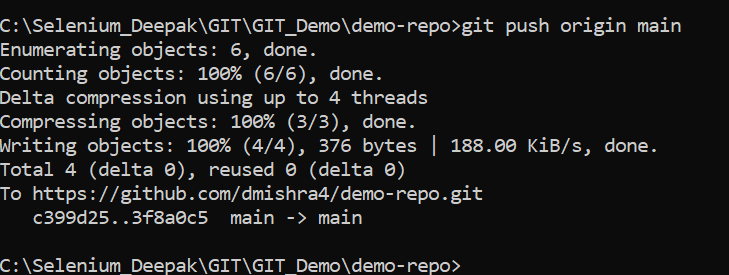
Command : git version -v



############## How to push code in GitHub ######################

First adding remote repository ( GitHUb) then execute below command

Command : git push origin main ( main is name of the branch)



##################### How to **add** GitHub Location ##############

Syntax : git remote add <shortname> <url>:

git remote add origin <https://github.com/dmishra4/demo-repo2.git>

Command : git remote [ it will list the short name for added remote repository]

git remote -v [ to display all added remote repository with URL used for reading and writing to that remote]

Then push the command to GitHub

Syntax :  git push <remote> <branch>

Git push origin master

If you want to **see more information** about a particular remote, you can use the git remote show <remote> command.

**Example**: git remove show origin

You can also change the remote repository short name by using below command :

Syntax: git remote **rename** <old name> <new name> [ here you are changing name from oringin to deepak ]

Example: git remote rename origin Deepak

You can also **delete** the added remote repository using below command .

Syntax: git remote remove origin or git remove rm origin

**Note**: origin — that is the default name Git gives to the server you cloned from.

git clone command implicitly adds the origin remote for you

############################# How to set Up stream #################

Git push **-u** origin master

Note : After executing above line , we do not need to write origin master as by default it will go there only

############## . GIT Branching #####################

Branch location :: .git\refs\heads

How to know how many branch are exist

**Command** : git branch ( it will display branch info in Git not in GitHub)

Commod : git branch -r ( it will display remote branch details)

Command : git branch -a ( To show all available branch in GIT and GITHUB)

Note : \* shows the current branch we are in.



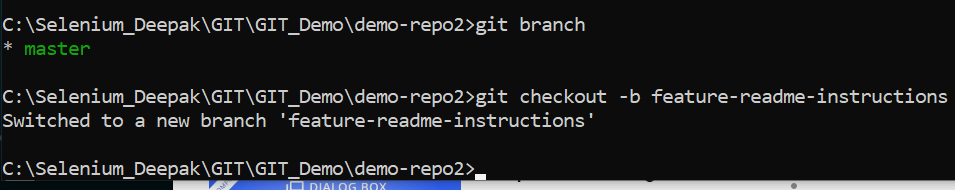
############# How to create/switch to new branch ###########

Command for Create branch : git branch branch\_name

Command to switch branch : git checkout branch\_name

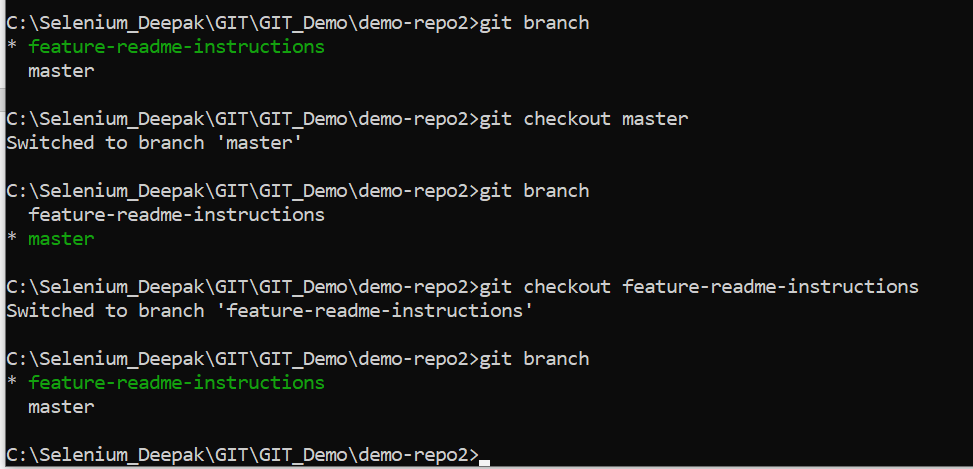
Command to create and switch to created branch at a time:

git checkout -b feature-readme-instructions



################## How to switch between branches ###########################

Command : git checkout branch-name ( Example : git checkout master )



###################### How to delete the Branch #################

Delete branch from local git : git branch -d <branch\_name> or git branch -D <branch\_name>

Example :

The -d option will allow you to delete the branch only if you have pushed and merged with remote branch and . Use -D option if you want to delete the branch even if you did not merged or push yet.

Delete branch from Remote Repository: git push <remote> --delete <branch\_name>

Example : git push origin -- delete feature

Try to synchronize branch using below command :

Command : git fetch -p

After fetching , branches which are no longer available on the remote will be deleted.

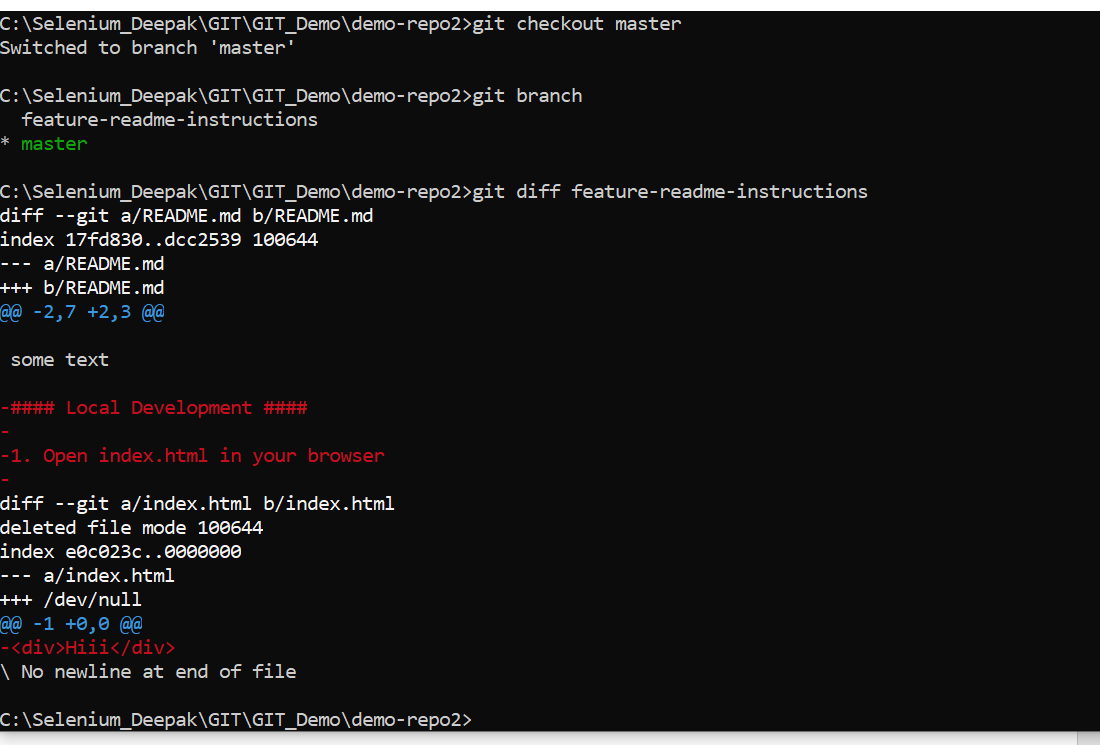
############### How to see difference between master and feature branch ###

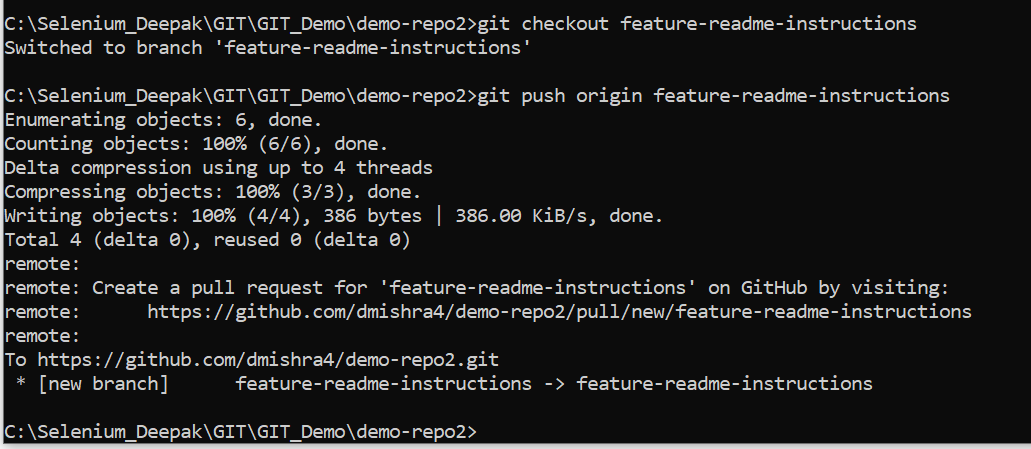
Firstly, go to master branch using command git checkout master and then type below command

Git diff new\_branch\_name ( git diff – feature-readme-instructions )

Then push the command to git hub

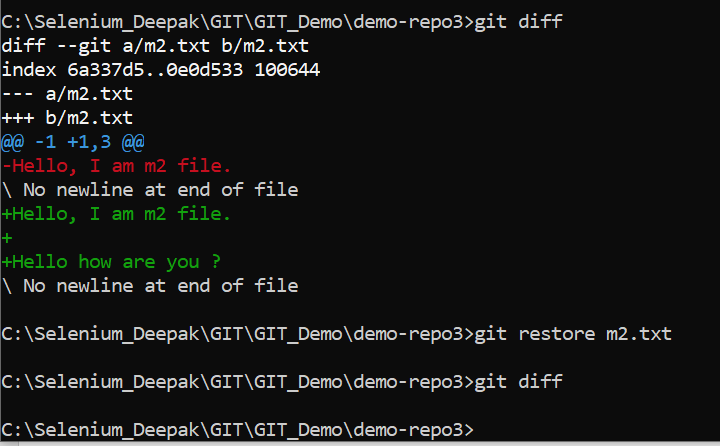
Git push origin feature-readme-instructions





####### How to see diff between working directory and Staging directory #######

Git command : git diff



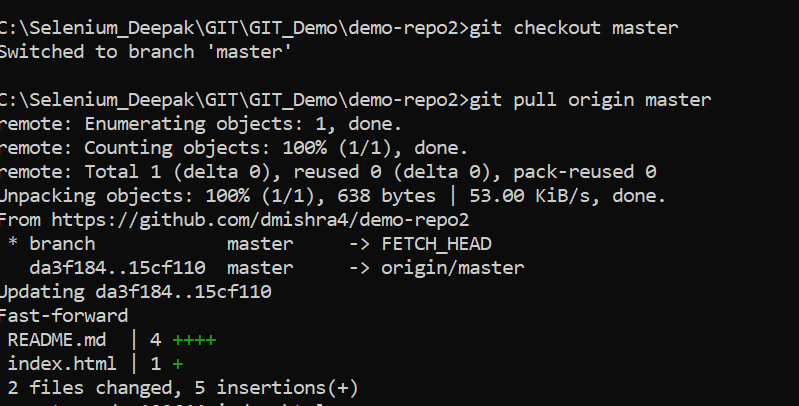
####### How to see diff between Staging directory and local repositoty#######

Command : git diff --staged

############################ How to Pull from GitHub ###############

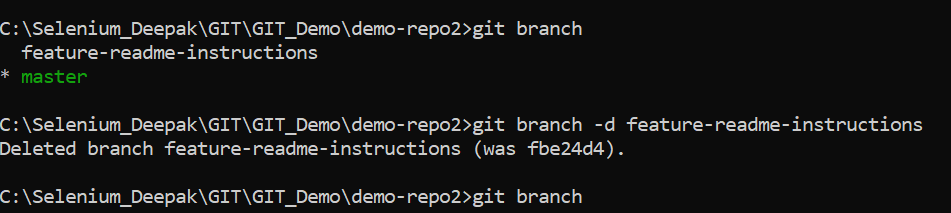
Syntax : git pull <remote> <branch>

Git pull origin master



############################### How to delete the branch ####################

Git branch -d branch-name



############################# .gitignore ###################################

First create .gitignore in root folder

Enter below line in that file :

\*.log

\*\*/target ( here target is a folder )

Then add this file in staging area using command : git add “.gitignore”

Then commit this in git using command : git commit -m “Added .gitignore file” -m “ added this file in root folder”

################ How to convert tracked file to untracked file in stating area #######

Suppose there is a tracked file in staging area called file.txt

Now to change the status from tracked to untracked, please write command

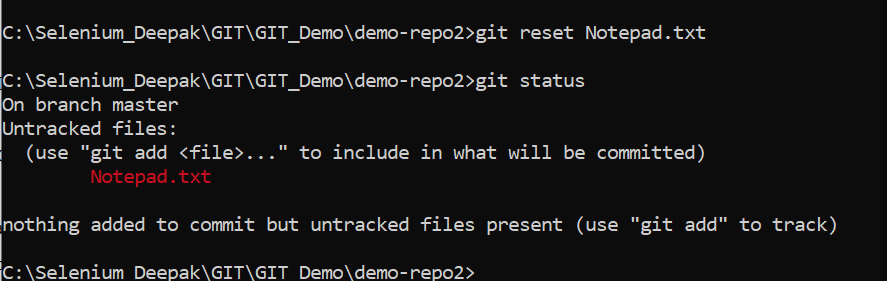
Git rm –cached index.html

########### How to send file from staging directory to working directory #################

**Git command:** git reset file\_name

To send all file to working directory :

**Git Command:** git reset



###############How to discard change in working directory #############

Command : git restore file\_name

############################# How to Merge code ################

First go to master branch then

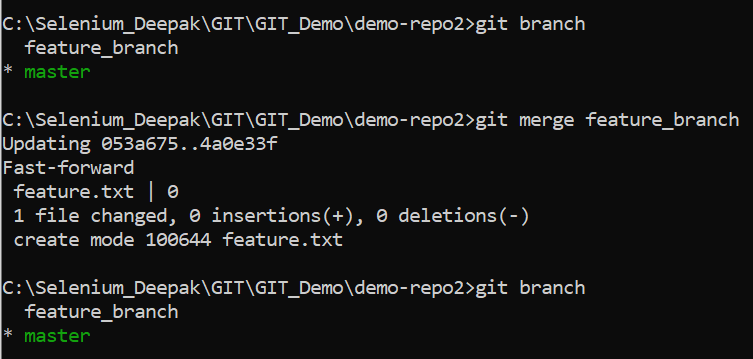
Merger Command : git merge branch\_name

Example : git merge feature\_branch

Now delete the branch after merging

Command : git branch -d branch\_name

Note : In case of conflict,  If you want to see which files are unmerged at any point after a merge conflict, you can run git status:



**This is a new README file  
  
<<<<<<< HEAD  
This is an edit on the master branch  
=======  
This is an edit on the branch  
>>>>>>> branch\_to\_create\_merge\_conflict**

As you can see, Git added some syntax including seven "less than" characters, **<<<<<<<** and seven "greater than" characters, **>>>>>>>**, separated by seven equal signs, **=======**. These can be searched using your editor to quickly find where edits need to be made.

That there are two sections within this block:

* The "less than" characters denote the current branch's edits (in this case, "HEAD," which is another word for your current branch), and the equal signs denote the end of the first section.
* The second section is where the edits are from the attempted merge; it starts with the equal signs and ends with the "greater than" signs.

A conflict arises when two separate branches have made edits to the same line in a file, or when a file has been deleted in one branch but edited in the other. Conflicts will most likely happen when working in a team environment.

####################### How to create tag ##################################

Git command : git tag v1.0 ( for local repository )

Git command : git push origin v1.0 ( for remote repository)

################### How to delete tag #######################

Git command: git tag -d v1.0 ( remove tag from git )

**Git command:** git push origin -d v1.0 ( remove tag from git hub)



################################ rebase #####################

Make sure you are in feature branch ( git checkout feature) then execute below command

Git command : git rebase master

**Git merge:**

* it is a non-destructive operation
* Existing branch are not changed in any way
* Creates a new merge commit in feature branch

**Git Rebase:**

* Moves the entire feature branch on the tip of master branch
* Re-writes the project history
* We get much cleaner and linear project history

############################# Undoing in GIT #########################

Sending file from staging to working directory.( Whenever you make any change in any file then first you have send that changes in staging area before the committing )

Command : git reset filename or git reset

How to see latest Commit ::

Command : git show HEAD

How to undo last commit ::

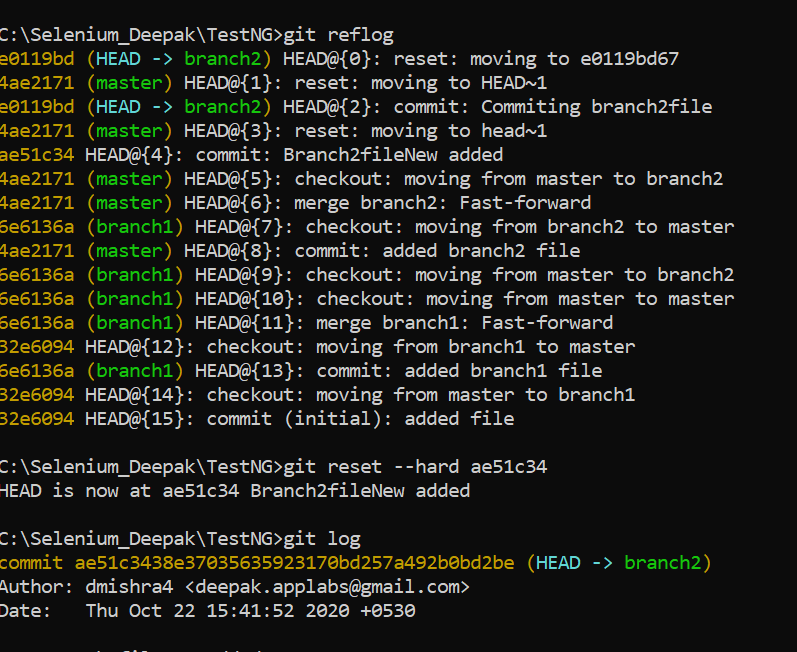
GIT Command : git reset HEAD~1 ( it will remove the last commit )

Then write git diff to see the difference.

######################### How to retrieve deleted commit ###############

Command : git reflog or git log -g

Then git reset --hard <SHA1Code > or git reset --hard HEAD@{0} or git reset --hard HEAD@{4}



############################### Fetch ########################

Fetch the value from remote repository

Syntax : git fetch <remote short name>

Example : git fetch origin

See the difference between local repository and the remote repository

Syntax : git diff <Local Branch> < Remote Branch>

Example : git diff master origin/master

Merge code after the fetch command

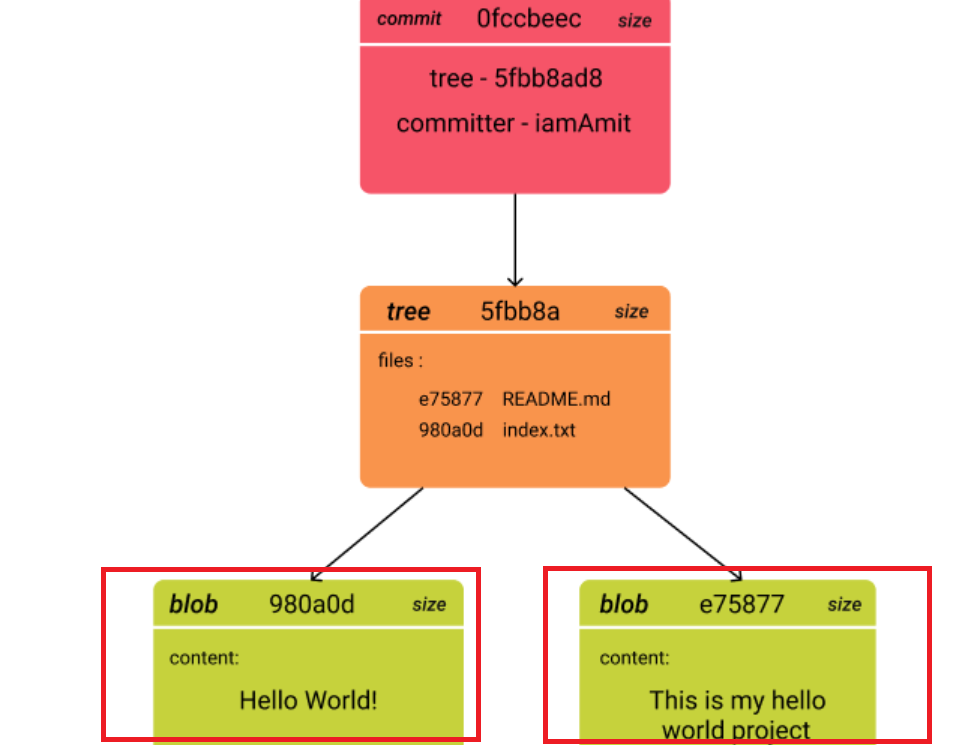
############################ GIT Internal ######################

Link : <https://medium.com/mindorks/what-is-git-object-model-6009c271ca66>

There are 4 object type in GIT which are **blob ( binary large object )**, **tree**, **commit** and **tag**.

Note : Whenever we execute the add command to track new file in git repository using below command, one blob object is getting created by GIT under .git/objects folder. The blob object stores the content of a tracked file including object type, size and hash value.

Command : git add <filename>



To see the content of any blob file : git cat-file -p <SHAVALue>

git cat-file -p 1953e56

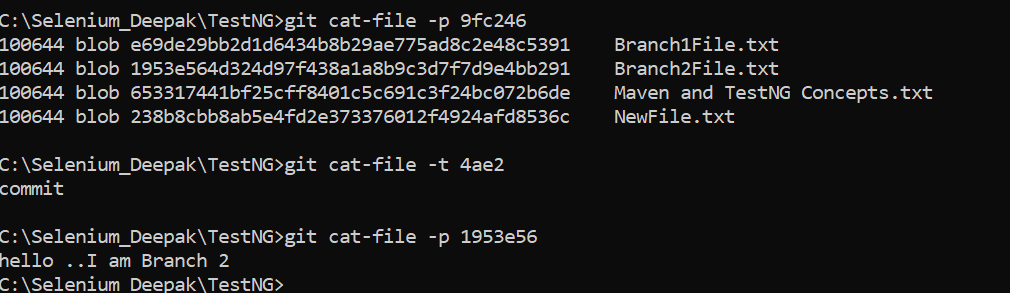
To see the object type, use below command ::

git cat-file -t 4ae2

To see the content of tree object

git cat-file -p 9fc246

Note : we can get hash value from the .git/objects folder



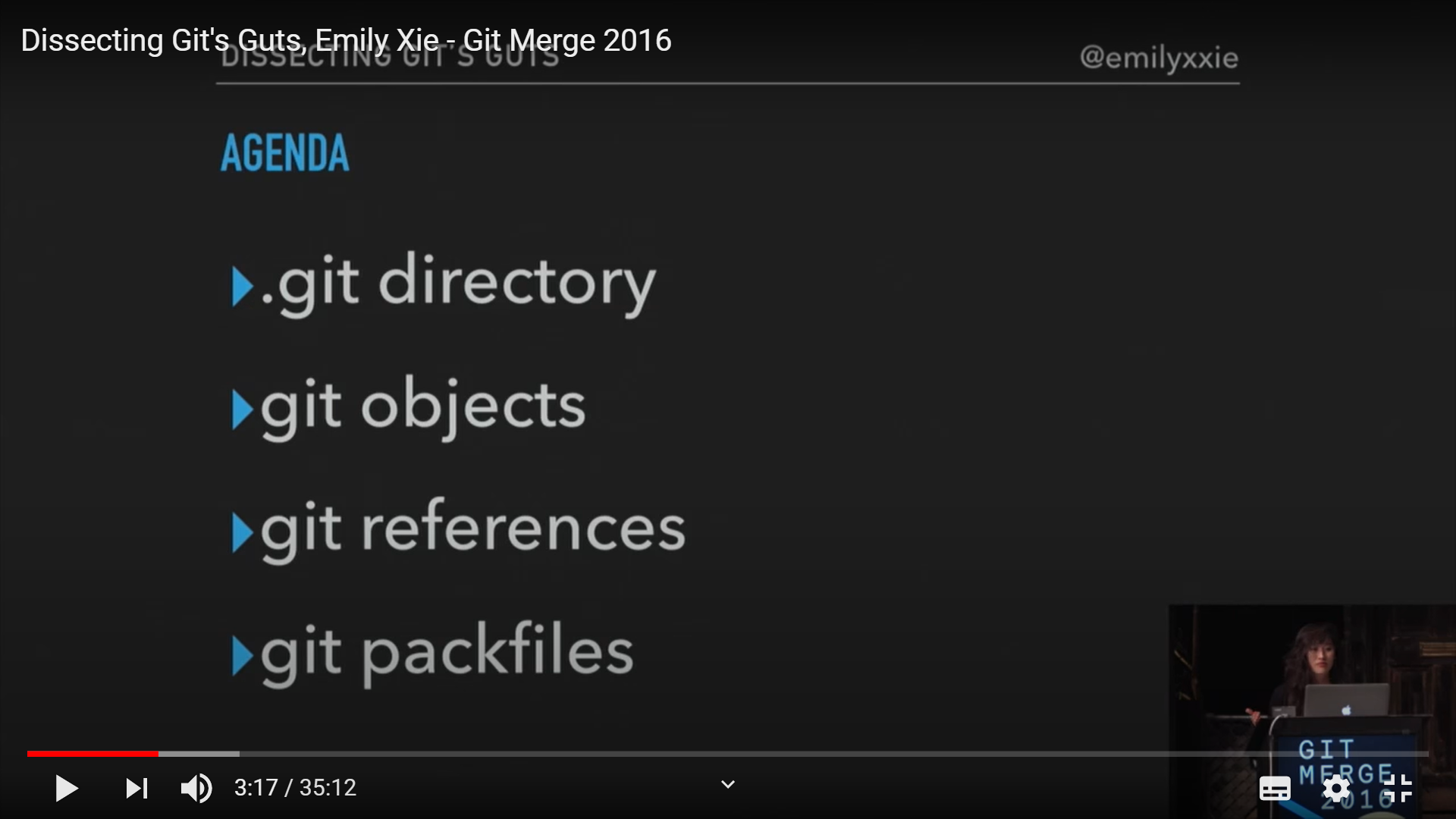
Once you perform the commit operation, two more git objects ( tree and commit ) will be created under .git/objects.

Tree object will contains list of all the created blob objects details including SHA1 , file name, type and rights. In addition, this information can be retrieve from below command

Command : git cat-file -p <SH1 value> ( See above screenshot to the the output)

To know the type of hash value , enter below command with requested HASH Value.

git cat-file -t 6e6136a134b7c7306b9b33821321bc8684125b22





Zlip : compression library git use

Use git cat-file -p command to read the content.

